<u>REMARKS</u>

This amendment responds to the Office Action mailed November 24, 2003. In the Office Action the Examiner:

- rejected claim 8 under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention;

- rejected claims 1, 3, 7, 9, 11, 15, 17, 19 and 23 under 35 U.S.C. 102(e) as being anticipated by Dobbs, U.S. Patent No. 6,039,426 (hereinafter Dobbs);

- rejected claims 2, 10 and 18 under 35 U.S.C. 103(a) as being unpatentable over Dobbs in view of Spencer, U.S. Patent No. 5,713,032 (hereinafter Spencer);

- rejected claims 4, 5, 12, 14, 20 and 22 under 35 U.S.C. 103(a) as being unpatentable over Dobbs in view of Koppolu, et al., U.S. Patent No. 6,268,924 (hereinafter Koppolu);

- rejected claims 6 and 21 under 35 U.S.C. 103(a) as being unpatentable over Dobbs in view of Weinberger, et al., U.S. Patent No. 5,644,682 (hereinafter Weinberger), in further view of Allen, et al., U.S. Patent No. 4,556,959 (hereinafter Allen), in further view of Mikkelsen, et al., U.S. Patent No. 5,995,724 (hereinafter Mikkelsen);

- rejected claims 8, 16 and 24 under 35 U.S.C. 103(a) as being unpatentable over Dobbs in view of Dennis, U.S. Patent No. 5,337,258 (hereinafter Dennis); and

- rejected claim 13 under 35 U.S.C. 103(a) as being unpatentable over Dobbs in view of Allen in further view of Mikkelsen.

With this amendment, Applicant has amended claim 15 to correct a typographical error. After entry of this amendment, the pending claims are: claims 1-24.

<u>Claim Rejections - 35 U.S.C. 112, second paragraph</u>

Applicant has amended claim 8 to replace the expression "executing said test engine" with the expression "processing said driver-test data structure". There is sufficient antecedent basis for this limitation in claim 1. Therefore, the rejections should be withdrawn.

<u>Claim rejections - 35 U.S.C. 102(e)</u>

Claim 1 recites a method of testing a print driver in a computer system. A print driver is a computer program that translates a document from a format that is not understood by a printer to another format that is acceptable to the printer. Since there are many document

formats, a print driver needs to be thoroughly tested to make sure that it translates documents from all different formats it supports to the acceptable format before being used in a computer system.

Specifically, given a print driver under test, the method first automatically generates a driver-test data structure that is associated with a plurality of applications and documents, and then processes the driver-test data structure to open the associated applications and documents and thereby test the print driver. A computer system illustrated in Fig. 2 of the present application shows that a print driver 56 and a driver-test data structure 72 are clearly two different objects stored in a computer memory 38. The print driver 56 is the target of the test and the driver-test data structure is a tool used in the test. In one embodiment, the driver-test data structure is a spreadsheet file. A test engine 70 opens the spreadsheet file and processes the information in the file row by row. The process includes executing an application, selecting one or more documents for the application automatically or through a user's guidance and submitting one or more printing jobs to the print driver under test (e.g., page 16, line 26 - page 17, line 9, page 18, line 28 - page 20, line 12, and Fig. 8).

In contrast, Dobbs teaches a method of determining for a particular media type which print mode of a printer's software or firmware driver will produce the highest print quality. In other words, given a print medium, e.g., a transparency, and an ink-jet print driver inherently having multiple print modes, Dobbs tries to find a print mode most suitable for the print medium. In the Office Action, the Examiner argues that the print driver acts as the driver-test data structure. Applicant respectfully disagrees.

As discussed above, the print driver and the driver-test data structure are two distinct components. In both Dobbs' patent and the present application, the print driver is the object or part of the object the respective methods try to improve. Dobbs' method tries to find a best print mode for the print driver and the print medium, while the present application tries to discover potential errors, if any, associated with the print driver using the driver-test data structure as a necessary tool.

Even if the term "print driver" is read broadly enough to include a data structure, a data structure that meets the requirements of claim 1 is not generated by Dobbs' method. Instead, Dobbs' method treats the print driver as a given and does not even suggest any effort to modify the driver in any way. Therefore, Dobbs does not teach the limitation of generating a driver-test data structure associated with a plurality of applications and documents recited in claim 1 (which includes generating a new or modified data structure). Nor does Dobbs

teach the limitation of processing the driver-test data structure to open the associated applications and documents. Column 2, line 56 of Dobbs cited by the Examiner simply suggests a way of operating the print driver to print a test pattern through an application program. Operating the Dobbs print driver in a very distinct process from having a test engine processes the driver-test data structure so as to test a print driver in conjunction with the applications and documents associated with the driver-test data structure.

Similarly, the icon on a computer screen (column 2, lines 54-57 of Dobbs) is not equivalent to the graphical interface recited in claim 3, because it does not allow the user to associate any application program with any document, as required by claim 3 and as shown in Fig. 8 of the present application.

In view of the foregoing, claim 1 and its dependent claims 3 and 7 are not anticipated by Dobbs. Meanwhile, claims 9, 11 and 15 are a set of computer program product claims, and claims 17, 19 and 23 are a set of computer system claims, both sets corresponding to claims 1, 3 and 7, respectively. Therefore, they are not anticipated by Dobbs for at least the same reasons discussed above.

Claim rejections - 35 U.S.C. 103(a)

To reject claims in an application under 35 U.S.C. § 103, the Examiner bears the initial burden of establishing a prima facie case of obviousness. *In re Bell*, 26 USPQ2d 1529, 1530 (Fed. Cir. 1993). In order to establish prima facie obviousness, the prior art, alone or in combination, must teach or suggest each and every limitation of the rejected claims. See *In re Vaeck*, 20 USPQ2d 1438 (Fed. Cir. 1991); *In re Royka and Martin* 180 USPQ 580 (C.C.P.A. 1974); and *In re Wilson* 165 USPQ 494 (C.C.P.A. 1970). The teaching or suggestion to make the claimed invention, as well as the reasonable expectation of success, must come from the prior art, not Applicants' disclosure. *In re Vaeck, Id.*

In the Office Action, besides Dobbs, the Examiner cited six additional references, contending that the combination of Dobbs and at least one of the six references renders at least one of the claims in the present application unpatentable.

However, for the reasons explained below, none of the six references cited by the Examiner teach or suggest the limitation of the driver-test data structure. Actually, none of them is related to the testing of a print driver. Therefore, Applicant respectfully traverses the rejections.

Spencer teaches a compound document processing system. The compound document refers to a document having more than one element, such as text, line drawings, images, sound sequences and/or motion sequences. However, Spencer does not teach or suggest anything related to the testing of a print driver. There is no mention of any print driver related data structure.

Koppolu teaches a document object with a print interface operating in a server application. In particular, the document object allows a client application to control the printing of a document through the print interface, such as progress monitoring and print cancellation. But Koppolu does not teach anything related to the testing of a print driver, and the print interface is simply a tool that **extends** the printing control from the server side to the client side.

Weinberger teaches a method for incorporating additional indicia, e.g., "CONFIDENTIAL", into a document image generated by a print driver without modifying the source of the document image, e.g., a WORD format document, such that the indicia can appear in a printed copy of the document. Weinberger does not teach or suggest anything related to the testing of a print driver. In contrast, the output of the print driver, a document image, is **directly** fed into a system that implements the method of incorporation without any modification.

Allen teaches a method of managing printing requests from multiple computers and redirecting each request to different printers according to the print options specified in the request in a network environment. Clearly, there is no mention of print driver testing in Allen.

Mikkelsen teaches a method of merging fixed and variable contents together to produce personalized documents. In particular, as shown in Fig. 3 of Mikkelsen, these fixed and variable contents have been processed by a print driver 52 **before** they are merged together by a raster image processor 58. Nowhere in Mikkelsen mentions anything related to the testing of the print driver.

Finally, Dennis teaches a method of measuring the execution time for a printer to draw various primitives, e.g., a line, a rectangle, or a circle, and then at run time, using such time cost metrics to determine if the printer can render the draw primitive in real-time or not. There is no mention of anything related to the testing of print driver.
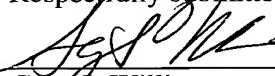
Since none of the cited references salvage the deficiency of Dobbs, namely, generating a driver-test data structure with associated applications and documents and

processing the driver-test data structure to open the associated applications and documents and thereby test the print driver, claims 2, 4-6, 8, 10, 12-14, 16, 18, 20-22 and 24 are patentable over the prior art of record.

In light of the above amendments and remarks, Applicants respectfully request that the Examiner reconsider this application with a view towards allowance. The Examiner is invited to call the undersigned attorney at 650-849-7721, if a telephone call could help resolve any remaining items.

Respectfully submitted,

Date: February 11, 2004

Gary S. Williams   31,066
Morgan, Lewis & Bockius LLP   (Reg. No.)
3300 Hillview Avenue
Palo Alto, California 94304
(650) 493-4935